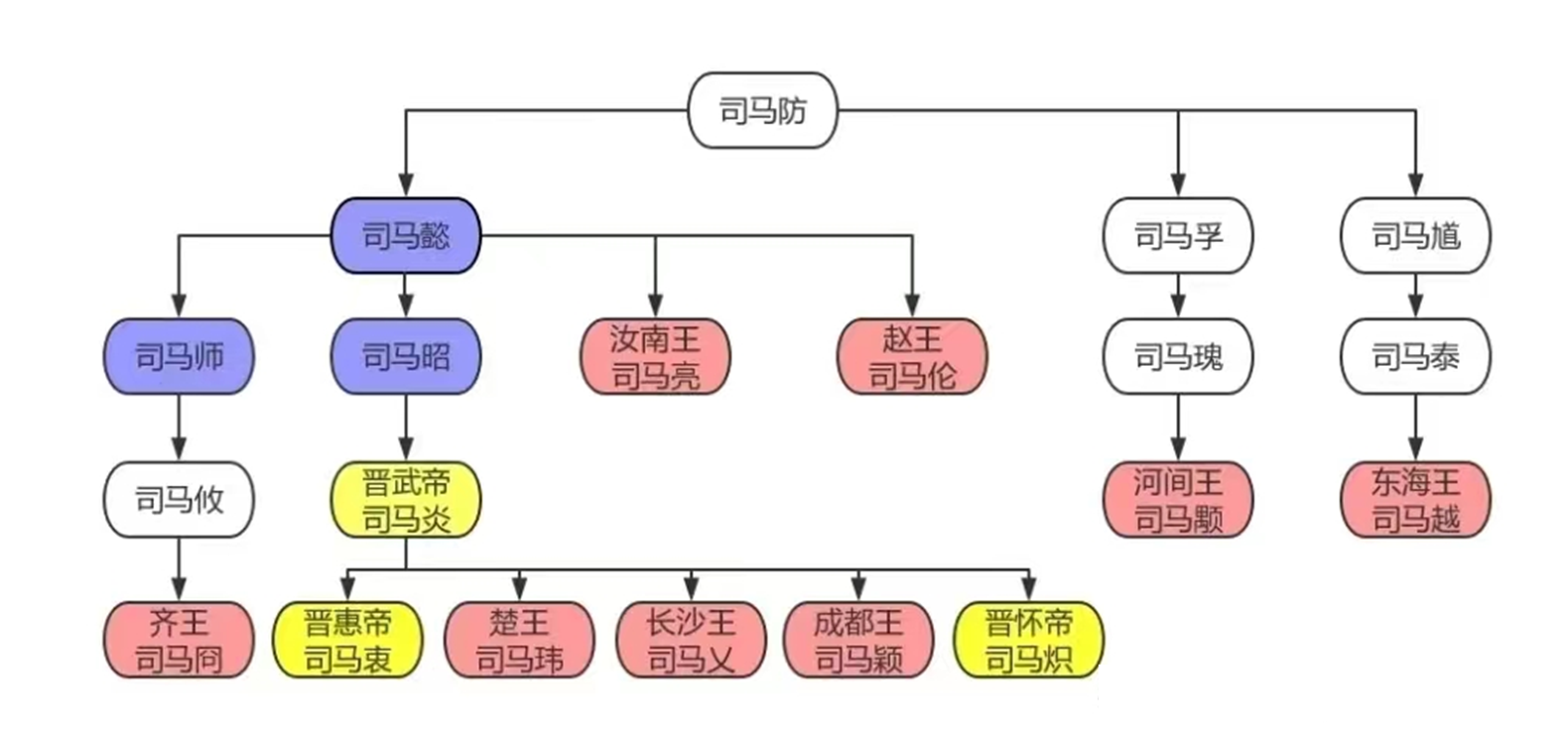
本次实习的目标是请同学们掌握一些典型的非关系数据类型在关系数据库中的处理手段。主要包括如下几方面的练习：

1. 递归查询
2. 时序数据的窗口查询
3. JSON和关系表的导入导出和基本查询
4. 向量数据库的使用

一、递归查询

下面是八王之乱的亲属关系示意图，我们先把它存入一个family表中，记录图中直接的父子关系。



create table family( father char(10), son char(10) );

insert into family values

(‘司马防’,’司马懿’),

(‘司马防’,’司马孚’),

(‘司马防’,’司马馗’),

(‘司马懿’,’司马师’),

(‘司马懿’,’司马昭’),

(‘司马懿’,’司马亮’),

(‘司马懿’,’司马伦’),

(‘司马孚’,’司马瑰’),

(‘司马馗’,’司马泰’),

(‘司马师’,’司马攸’),

(‘司马昭’,’司马炎’),

(‘司马泰’,’司马越’),

(‘司马攸’,’司马囧’),

(‘司马炎’,’司马衷’),

(‘司马炎’,’司马玮’),

(‘司马炎’,’司马乂’),

(‘司马炎’,’司马颖’),

(‘司马炎’,’司马炽’);

我们定义如下规则：

brother(X,Y):-father(Z,X),father(Z,Y). 有共同父亲的是兄弟。

ancestor(X,Y):-father(X,Y). 父亲是祖先。

ancestor(X,Y):-father(X,Z),ancestor(Z,Y).父亲的祖先是祖先。

请同学们完成如下SQL查询：

1. 找出所有的兄弟关系。
2. 使用递归查询找出所有祖先关系。
3. 自定义“叔侄”或“堂兄弟”关系的规则，然后使用SQL完成。

二、窗口查询

窗口函数是一项用途广泛的重要SQL功能，被广泛地应用于数据分析场合，同时也是数据岗的重点面试题型。我们的这次实习是基于窗口函数实现股票的量化因子和交易指标，二者选一即可。

1. 如果你选择实现量化因子，量化因子的类型以及公式定义是非常多的，可以参考这篇文章：

“101 Formulaic Alphas.pdf ”

以及[预测股票市场的101个alpha因子的解读与总结 - 知乎 (zhihu.com)](https://zhuanlan.zhihu.com/p/671078106)

https://zhuanlan.zhihu.com/p/671078106

我们的实习是实现上面pdf里面提到的101量化因子，同学们可以把这101个公式分组，1~25, 26~50, 51~75, 76~101，每组里面选择一个公式来用SQL的滑动窗口功能来实现（）。有兴趣的同学可以看看这些因子选出来的股票表现如何。

因子公式中的一些特殊参数：

* vwap：成交量的加权平均价格，将当日成交总值除以总成交量即可
* rank(x) = cross-sectional rank：表示某股票x值在横截面上的升序排名序号，并将排名归一到[0,1]的闭区间

1. 如果你选择实现交易指标，请调研MACD、BOLL、KDJ这三个经典指标的计算公式，选择其中的两个来用SQL来实现它们。

三、JSON操作

JSON是比较重要的一种数据类型，我们本次实习的目标是掌握JSON的一些基本概念和处理技巧，包括如何把关系表数据和JSON的相互转换，以及对JSON的基本查询。

1. 基于平台上的订单数据库，生成一个描述customer的JSON对象，这个对象结构的含义如下： customerId和customerName是键值对；addr是个键值对，它的值又是一个键值对集合，包括country和city；一个customer购买的所有product放在一个数组中，数组中每个元素又是一个键值对，记录所购买的每种product的总金额（如果购买的product数量过多，取前5个即可）。（这部分功能非常实用，因为前端处理的对象模型通常都以JSON形式声明，所以需要完成从后端数据库到JSON的映射。一般会用到group\_concat，以及json\_object, json\_array函数来完成。）

{ customerId: 20,

customerName: 'Kane, John',

addr:{country: Austria, city: ‘Graz’ },

[productname:total\_money,…] }

1. 将上面生成的JSON对象导入一个关系表custInfo(id int, Info json)中，完成如下查询任务：
2. 指定一个country，返回位于该country的customer。（典型的路径查询，比较简单）
3. 计算每种product的总金额。（有一定难度，需要把数组展开）

四、向量数据库实习设计

本次实习设计的目的是借助自然语言处理中的典型任务，体验一个非结构化数据的典型处理流程。大概处理过程如下：

1. 建立小说表 yttlj ( phaseId, phaseText );（平台提供）
2. 建立人物表 person( personId, personName );（平台提供）

为该表添加一个字段，统计每个人物的词频，也即他在多少段落中出现过。有兴趣的同学可以尝试用Python中的词云工具来显示一下。

1. 建立共现表 coCurrence( personId1, personId2, counts );

（这部分请同学们用SQL实现，一个思路是在person表中增加一个类似位向量的字段phaseContains，每个位对应一个段落，如果某段落包含了该person，则将该位设置为1，这样通过两个将person的phaseContains进行位与运算，即可得到他们的共现次数。另一种做法是生成人物对(person1,person2), 针对每对人物，看看有多少段落同时包含了这两个人物）；

1. 通过矩阵分解，得到词向量表示，这部分可以通过平台提供的Python代码实现。
2. 将上面的词向量存入pgvector，计算人物之间位向量相似度，找到每个人和他最相似的另一个人。